Approximating TSP Solutions with Graph Pyramids

Walter G. Kropatsch and Yll Haxhimusa*
Institute of Computer Aided Automation 183/2
Vienna University of Technology
Pattern Recognition and Image Processing Group

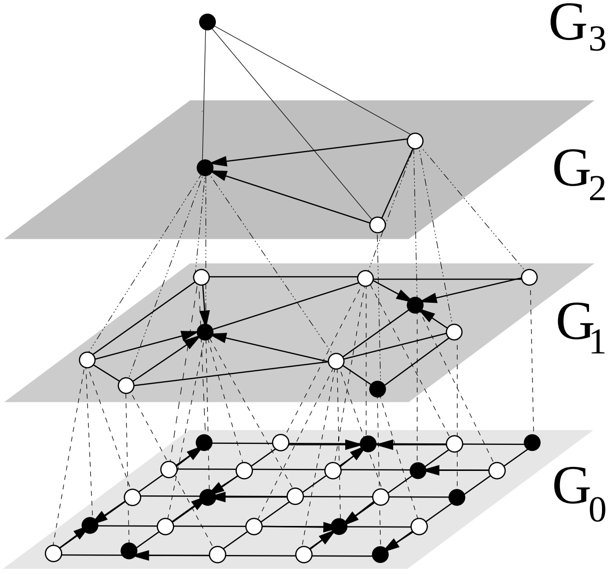
CONTENTS:

- Image pyramids with graphs ...
- ...for Image Partitioning ...
- ...with Minimal Spanning Tree (MST)
- TSP in 1D
- simple and difficult solutions
- a too simple Algorithm
- Problems with identical solutions

*Supported by the Austrian Science Foundation under grant FSP-S9103-N04.

Image Pyramids

- Hierarchical structures - Pyramids,
- Properties of Pyramids:
  - Structure,
    - ∗ horizontal and vertical relations
  - Content of the cells,
    - ∗ numeric, symbolic or both
  - Processing of a cells



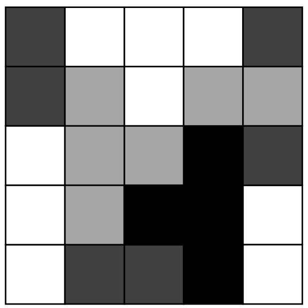
Properties of Image Pyramid

- Regular image pyramid
  - $log()$ height $\Leftrightarrow$ constant reduction factor
  - Lack of shift invariance
  - Confined to globally defined sampling grid
- Irregular image pyramid
  - Biological systems, e.g. human retina not regular
  - Perturbations may destroy the regularity of regular pyramids
  - In general not $log()$ height

See book of Jolion and Rosenfeld [5] for more details

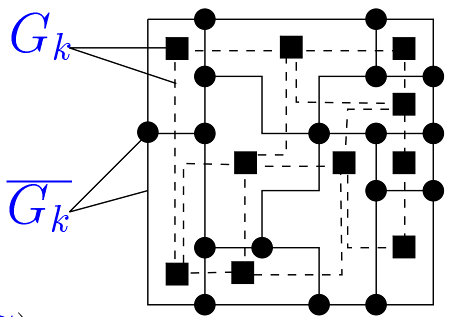
Irregular Graph Pyramid

- Planar connected attributed graphs $(G_k,\ \overline{G_k})$
- Pyramid is a sequence of $(G_k,\ \overline{G_k})$, $0 \leq k \leq h$
- Dual graph contraction (DGC) [6]



Image

$\leftrightarrow$
Graph
contraction (DGC)



Dual graphs

Bottom-Up Construction by Dual Graph Contraction

Input: Graph $G = (V, E)$ and its dual graph $\overline{G} = (F, \overline{E})$

1. **while { further abstraction is possible } do**
   (a) select contraction kernels $CK \subset E$
   (b) dual edge contraction G/CK and
   (c) simplification of dual graph $\overline{G/CK} \setminus \overline{SK}$,
   (d) apply reduction functions to compute content of new reduced level.

Output: Irregular graph pyramid

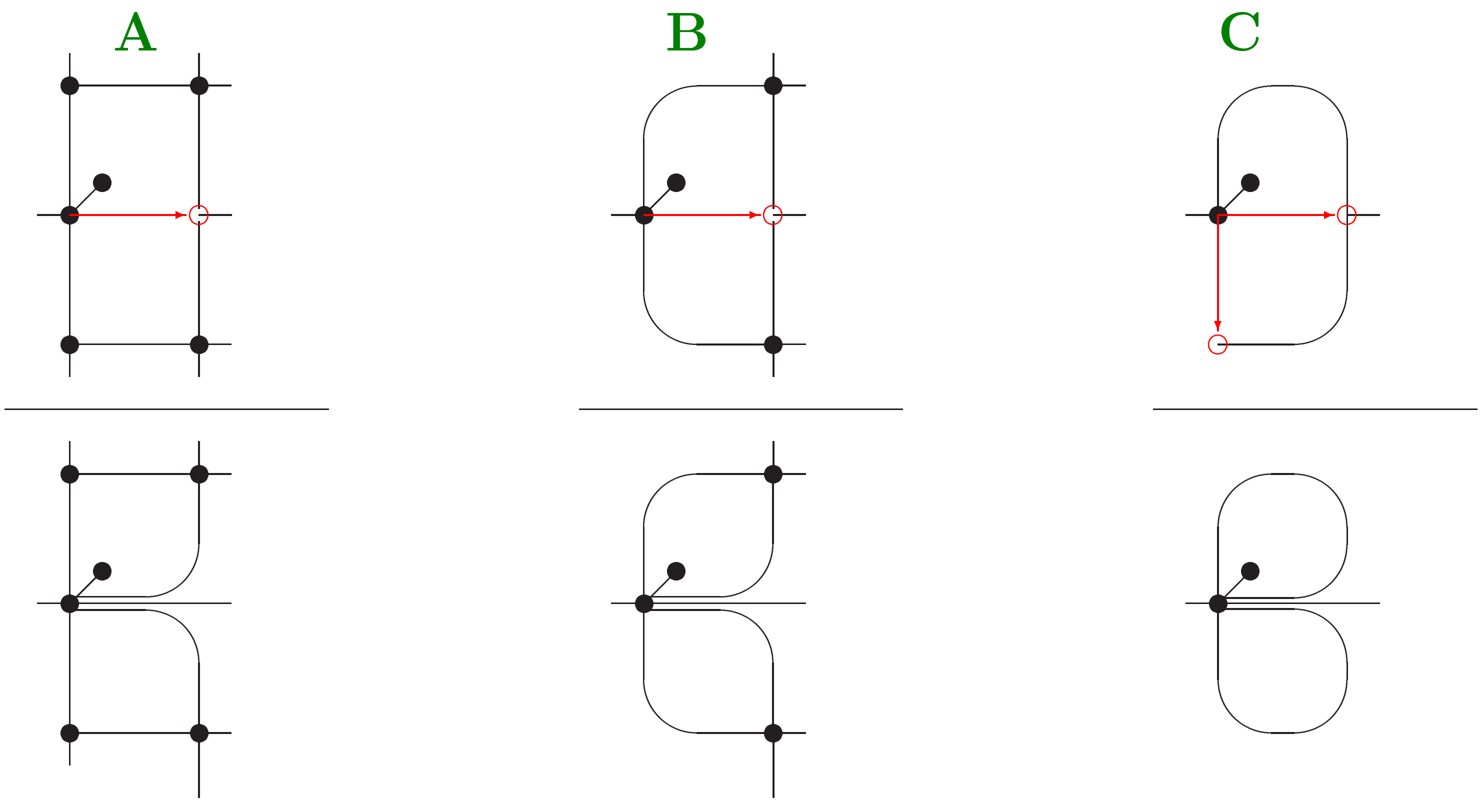
Simplification kernel SK removes redundant self-loops and multi-edges.
Content influences steps (a) and (d);
Operation is purely structural!

Dual Graph Contraction [6]: (1) Edge Contraction

3 Cases:

$G_k(V, E)$
$K_{k,k+1}$

$G/\{e\}$



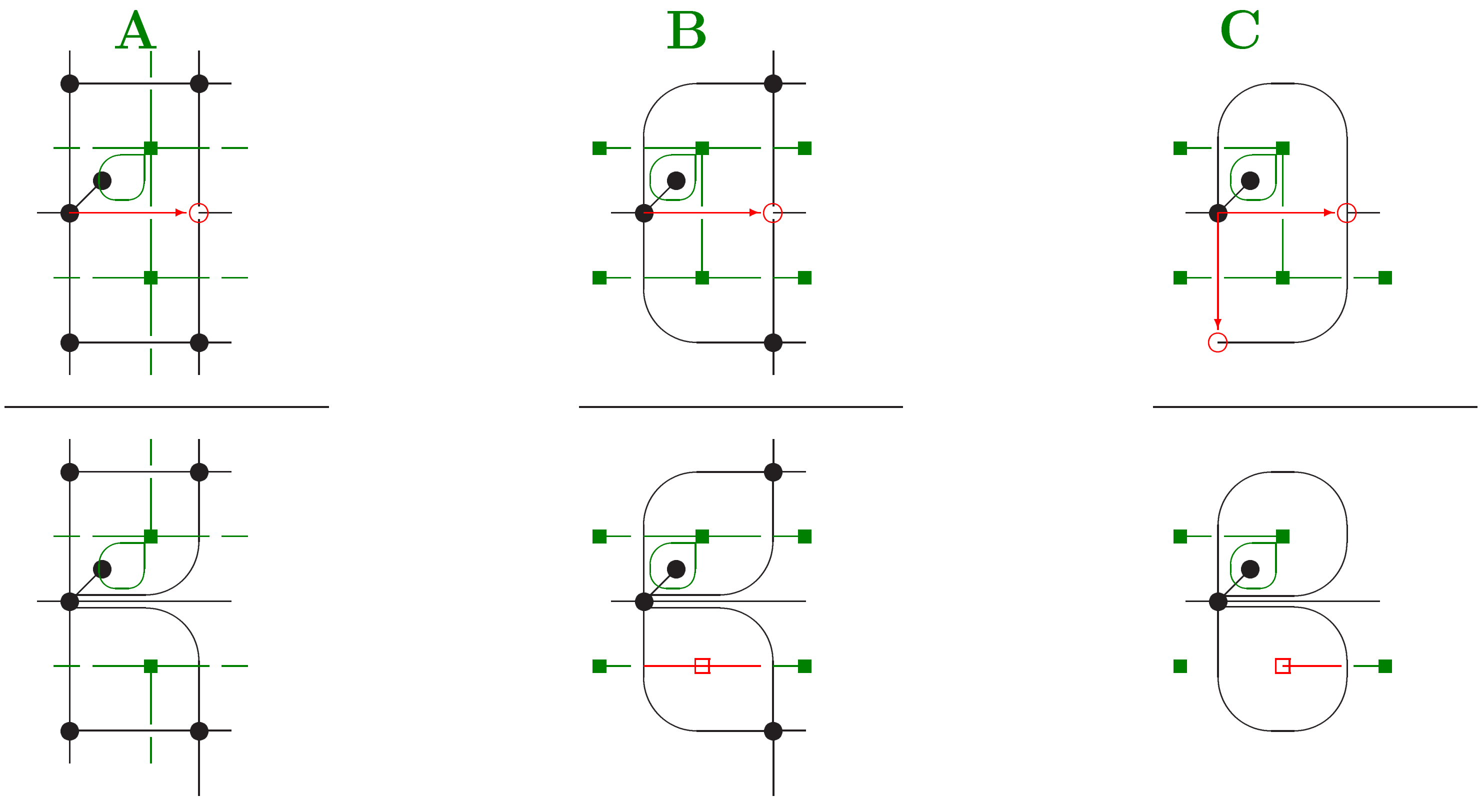
... preserves the connectivity, but can produce multiple edges and self-loops

(2) Simplification

3 Cases:

$(G, \bar{G})$

$\bar{G} \setminus \{e\}$



Edge contraction $\Leftrightarrow$ Edge removal in dual graph

Multiple edges and self loops $\Leftrightarrow$ Vertices of degree 2 and 1 in $\bar{G}$.

Dual Graph Contraction Summary

Level	representation	contract / remove	conditions
0	$(G_0, \overline{G_0})$		
	$\downarrow$	contraction kernel $K_{0,1}$	forest, depth 1
	$(G_0/K_{0,1}, \overline{G_0} \setminus \overline{K_{0,1}})$		
	$\downarrow$	redundant multi-edges, self-loops	deg $\overline{v} \leq 2$
1	$(G_1, \overline{G_1})$		
	$\downarrow$	contraction kernel $K_{1,2}$	forest, depth 1
	$\vdots$		

Image Partitioning by Graph Pyramids

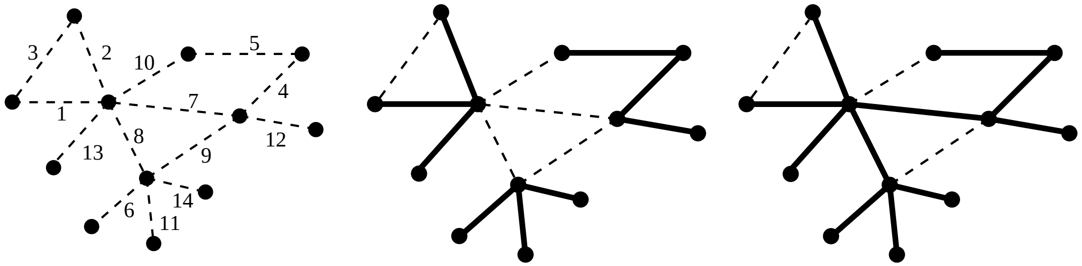
- Low level cue image segmentation cannot produce a final "good" segmentation.
- Grouping method should have the following [4]:
  - create a hierarchy [10],
    - graph pyramids
  - capture perceptualy important groupings,
    - internal and external contrast
  - run in linear time,
    - Minimum Spanning Tree (MST) based algorithm.

Minimum Spanning Tree

- Graph $G(V, E, w)$ connected and attributed by weight w
  - $w : e \in E \to R^+$
- Goal : Find the spanning tree T with the smallest weight $\sum_{e \in T} w(e) \to \min.$
  - Kruskal's algorithm [7]
  - Prim's algorithms [9]
  - Borůvka's algorithm [2]

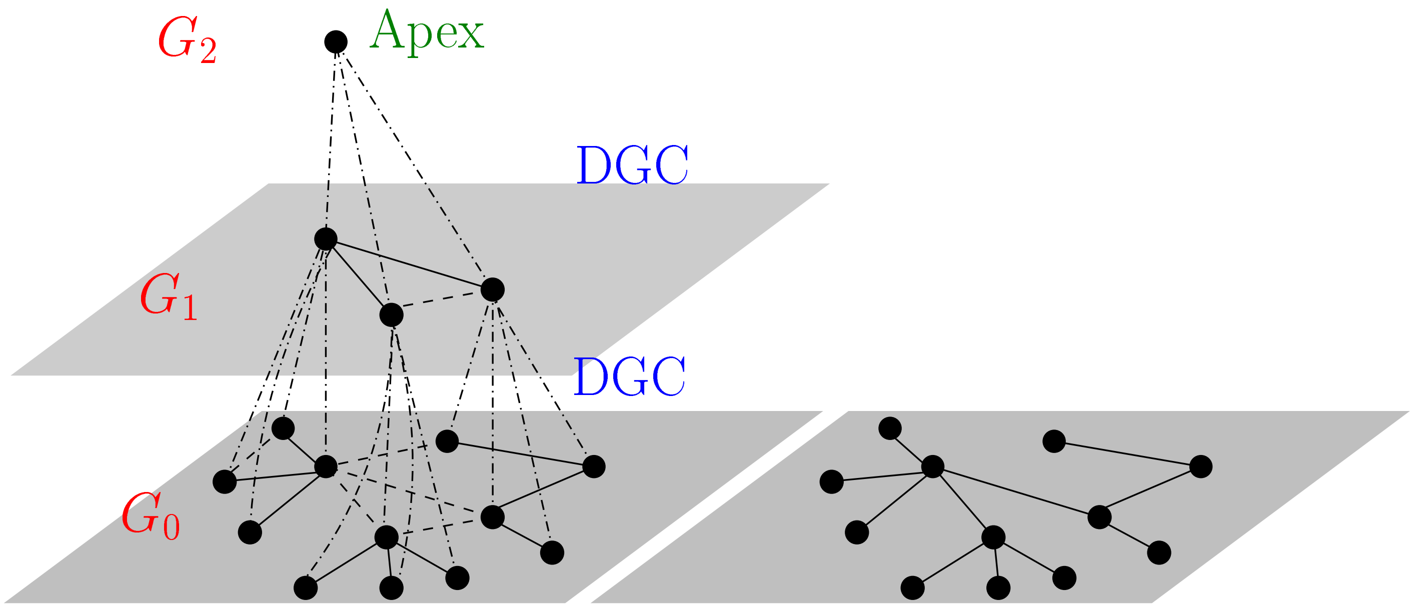
Borůvka's Algorithm [2]

- Input: Graph $G(V, E)$.
  1. $MST :=$ empty edge list.
  2. $\forall v \in V$ make a list of trees L.
  3. **while** {there is more than one tree in L} **do**
     - each tree $T \in L$ finds the edge e with the minimum weight which connects T to $G \setminus T$ and add edge e to MST.
     - edge e merges pairs of trees in L.
- Output: Minimum weight spanning tree - MST.



Borůvka's Algorithm and Dual Graph Pyramid

- dual graph contraction (DGC) contracts all trees $T \in L$ in step 3.



$MST(apex) = K_{0,2}$

Some Results: Hierarchies

Ramp: size= 223×110; $\tau = 1000$

Level 0 (24 753) Level 8 (44) Level 9 (25) Level 10 (13) Level 14 (2)

Woman: size= 116×261; $\tau = 300$

Level 0 (30 276) Level 10 (86) Level 12 (31) Level 14 (8) Level 16 (3)

Level k ($\#|CC|$)

Some Results: Hierarchies, cont.

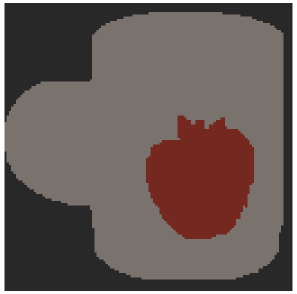
Object45: size= 128×128; $\tau = 300$


Level 0 (16 384)


Level 8 (129)

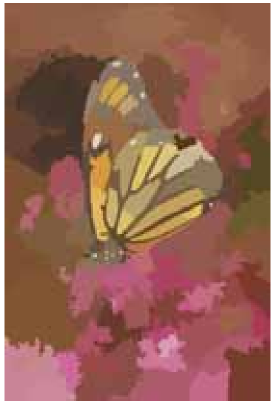

Level 10 (43)


Level 12 (13)


Level 14 (3)

Monarch: size= 768×512; $\tau = 300$


Level 0 (393 216)

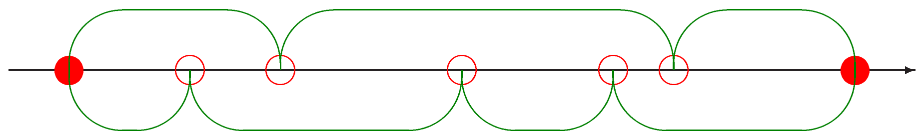

Level 14 (108)


Level 16 (57)

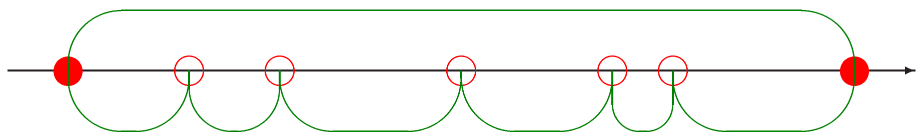

Level 20 (25)


Level 22 (18)

TSP in 1D



- in 1D: Cities are ordered
- $\exists x_{min}, x_{max}$
- length of circuit $= 2(x_{max} - x_{min})$

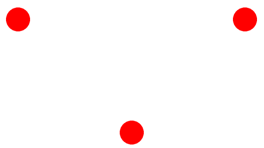


- other solution exists:
- For n cities 2^{n-2} **solutions** exist!

"Simple TSP-Configurations" in 2D

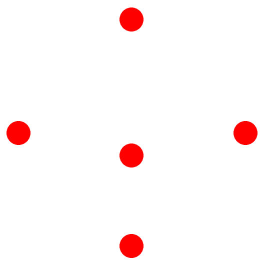
- All cities on a line = 1D Problem
  $ax_i + by_i = c$ for some $a, b, c \in \mathcal{R}$
- $n = 3$ triangle is trivial.
- Are large number of cities DIFFICULT?
- not always, e.g.:

Triangle



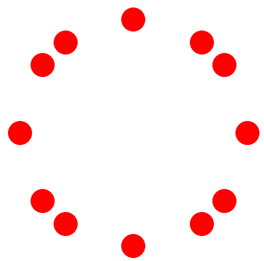
TRIVIAL

5 Cities



DIFFICULT

12 Cities



SIMPLE

A Simple Algorithm

Given: n cities $C_i, i = 1, \ldots, n$

1. find start city $T_1 = C_i$, k=2
2. **while {** $\exists$ city $Q = C_j$ to visit **} do**
   connect T_{k-1} to next closest city $T_k = Q$, k=k+1

+ complexity $\mathcal{O}(n*\text{search for closest city})$

– does not always find the best tour

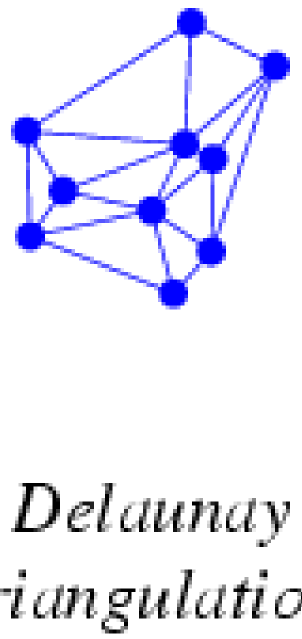
+ but sometimes succeeds

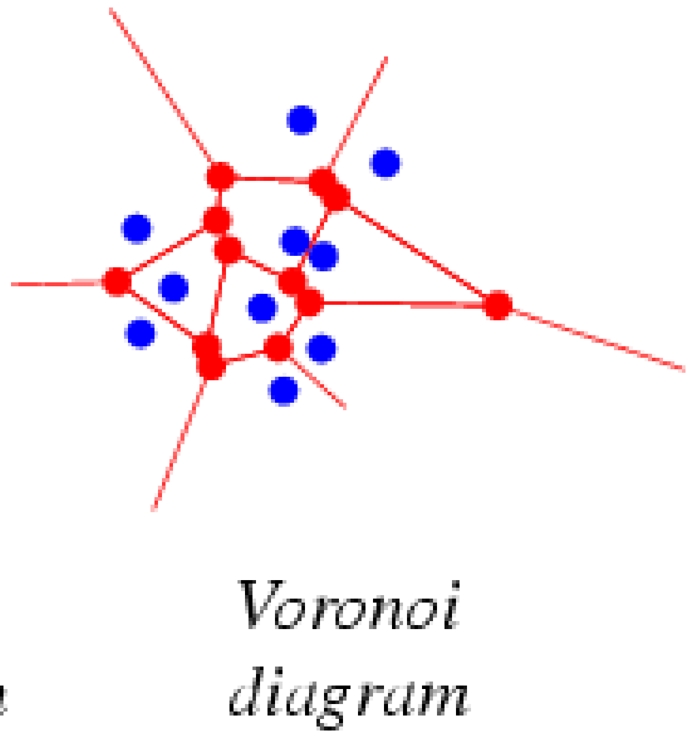
? when? How often?

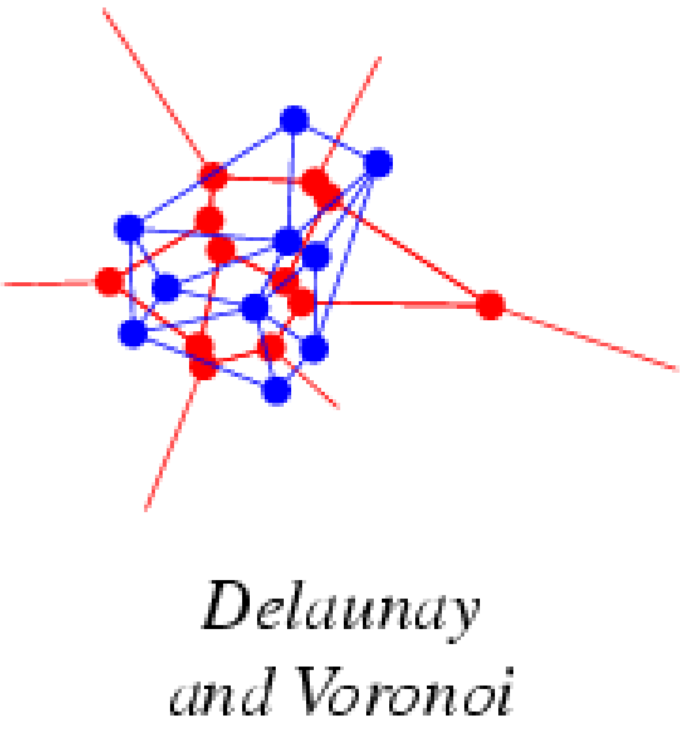
How to Organize/Partition City-Space?

1. Raster cell with/without city
2. Graph $G(V, E)$: city = vertex $v \in V$; edges $e \in E$?
3. complete graph: $E = V \times V$
4. Delaunay triangulation $E \subset V \times V$
5. Voronoi Diagram

Voronoi Diagramm, Delaunay Triangulation







Voronoi Diagram, Time and Space Complexity

The Voronoi Diagram is the dual of the Delaunay Triangulation.

- Time: $O(n \log n)$
- Space $O(n)$

Lower bound for computing Voronoi diagram is $\Omega(n \log n)$,
for special cases it is linear in time $O(n)$ [1],
e.g. when the sites (points) are on the vertices of a convex polygon
Additional properties help to reduce the complexity of the problem.

Pyramid: Reduce Resolution

1. larger raster cells contain clusters
2. preserve approximate location
3. reduce number of cities
4. repeat until solution becomes trivial
5. refine solution top down to the base level

Graph Pyramid: Reduce Number of Edges

Number of cities $|V| = n$;
number of edges $|E|$ varies;
embedding in the plane has faces F.
Related by Euler formula: $|V| - |E| + |F| = 1$

$G(V,E)$	$\|E\|$	$\|F\|$	comment
complete graph	$\binom{n}{2}$	–	contains TSP solution
Triangulation	$3(n-1)-b$	$2(n-1)-b$	$3 \leq b \leq n$ edges on boundary
MST(G)	$n-1$	0	sum of edges minimal
triangulated TSP	$2n-3$	$n-2$	is the goal (b=n)

TSP with triangle inequality

Fakts:

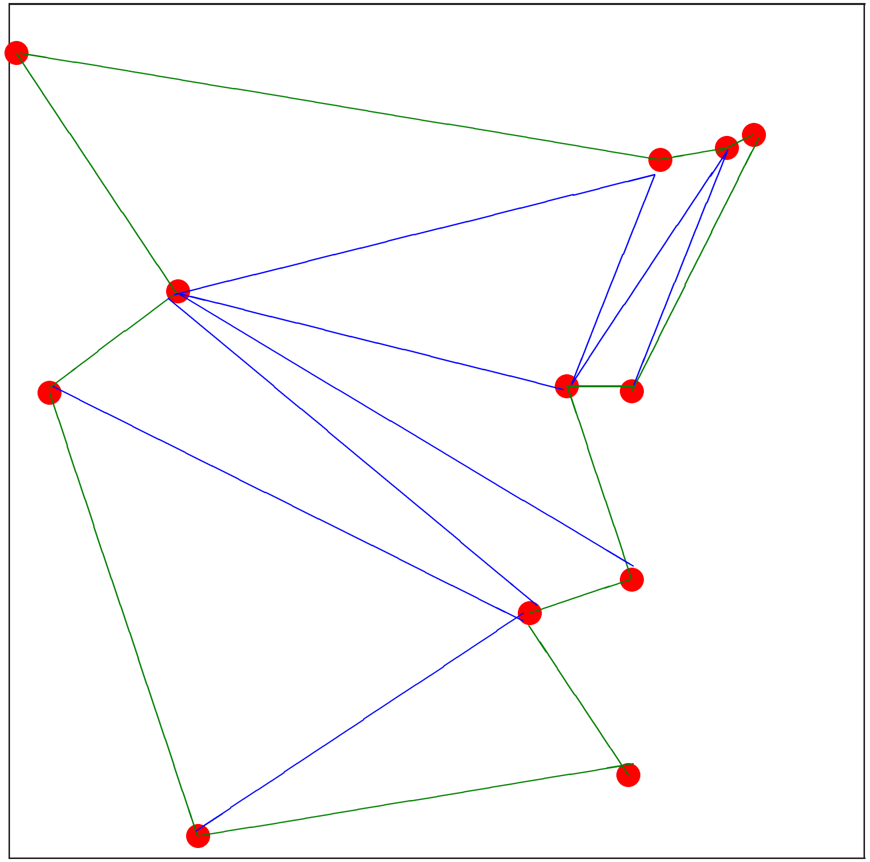
- TSP with triangle inequality: That is, for any 3 cities A, B and C, the distance between A and C must be at most the distance from A to B plus the distance from B to C. Most natural instances of TSP satisfy this constraint.
- MST is a natural **lower bound** for the length of the optimal route.
- In TSP with triangle inequality, it is possible to prove **upper bounds** in terms of the minimun spanning tree → 'Christofides Heuristics' …

Christofides' Heuristics [3]

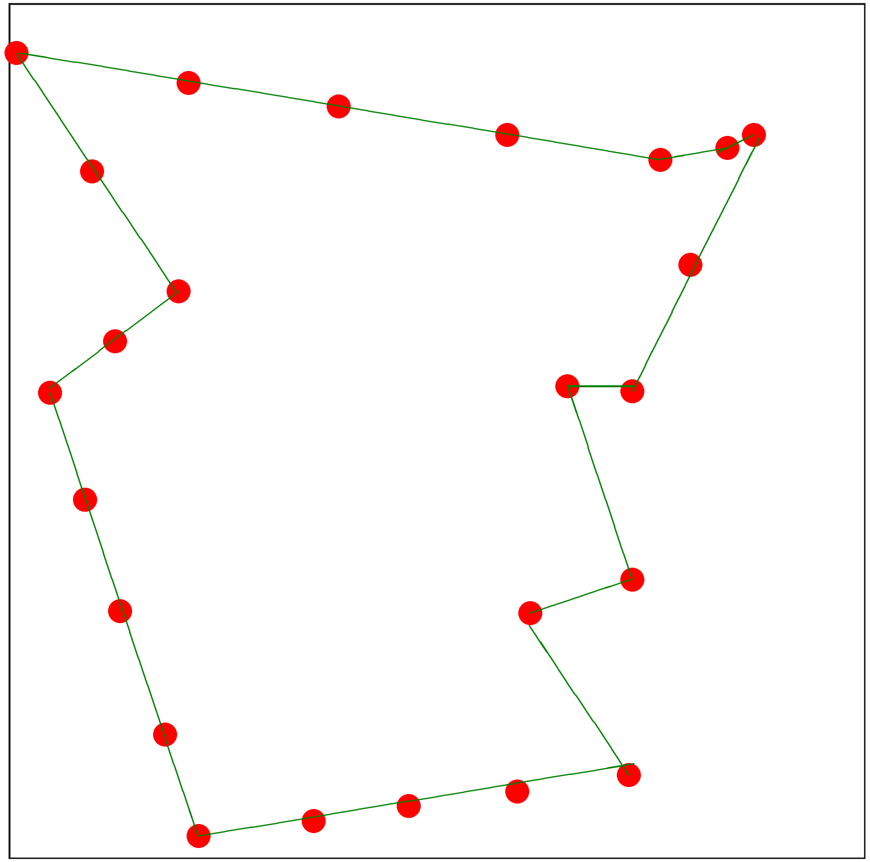
- Construct the minimal spanning tree T
- Find the perfect matching M among vertices with odd degree
- Combine the edges of M and T to make a multigraph G
- Find an Euler cycle in G by skipping vertices already seen

Christofides' algorithm combines the minimum spanning tree with a solution of minimum-weight perfect matching.
This gives a TSP tour which is at most 1.5 times longer than the optimal tour.
It is known, however, that there is no polynomial time algorithm that finds a tour of length at most $1 + \frac{1}{219}$ times the optimal, unless P=NP [8].

TSP-Configurations with the same tour length



$|V| = 12, |E| = 21, |F| = 10$



$|V| = 24, |E| = 45, |F| = 22$

Inserting Additional Cities

- . . . along solutions does not change length of solution.
- . . . along solutions does not change order of cities.
- after insertion the simple algorithm finds solution more often
- Is there a sampling theorem for distances along the solution?
- What is the number of problems that the simple algorithm can solve optimally?
- $|$simple problems$| > |$difficult problems$|$?
- Can we characterize and recognize simple problems?

References

[1] A. Aggrawal, L. J. Guibas, J. Saxe, and P.W. Shor. A Linear-Time Algorithm for Computing the Voronoi Diagram of a Convex Polygon. *Discrete & Comput. Geometry*, 4(6):591–604, 1989.

[2] Otakar Borůvka. O jistém problému minimálnim. *Práce Mor. Přírodvěd. Spol. v Brně (Acta Societ. Scienc. Natur. Moravicae)*, 3(3):37–58, 1926.

[3] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, 1976.

[4] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Image Segmentation Using Local Variation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–104, June 1998.

[5] Jean-Michel Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision*. Kluwer, 1994.

[6] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366–374, December 1995.

[7] J. B. Jr. Kruskal. On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem. In *Proc. Am. Math. Soc.*, volume 7, pages 48–50, 1956.

[8] C. H. Papadimitriou and S. Vempala. On the approximability of the traveling salesman problem). In *Proceedings of STOC'2000*, page extended abstract, 2000.

[9] R. C. Prim. Shortest connection networks and some generalizations. *Bell Sys. Tech. J.*, 36:1389–1401, 1957.

[10] Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation. In *Proceedings IEEE Conference Computer Vision and Pattern*, pages 731–737, 1997.